

Mobile Agents: Objective, Platforms and Architecture (The Cutting Edge of Wireless Technology)

Sandeep Srivastava¹, Meenakshi Arora²

Assistant Professor, Department of MCA, G.L. BAJAJ Institute of Technology & Management, Greater Noida, India

Abstract: *In the past few years the computer systems have evolved from monolithic computing device to much complex client-server environment, now new phase of evaluation allows complete mobility of application code among supporting platforms to form a loosely-coupled distributed system. Mobile agent paradigm is one such technology which has numerous applications where it can be beneficial to name a few areas where the mobile agents have potential deployment is database search, distributed systems and e-commerce. It can produce very good results in limited resources or in deficient environment where bandwidth and memory are significant constraint. But still it is not widely accepted due to its security issues. This work seeks to explain the objectives and properties of the mobile agents in currently used architecture and platform of mobile world from the currently used approach RPC (Remote Procedure Calling) and new approach RP (Remote Programming) of the mobile network we can distinguish the various used unused and new features of mobile agents. The paper tracks the various hurdles and properties used from the beginning of mobile world till now, lists out the advantages and shortcomings and explore architecture of mobile agent platforms that can help to remove those shortcomings.*

Keywords: Mobile Computing, Mobile Agents, RPC, RP, Mobile Agents Applications, Mobile Agents Properties

1. Introduction

Mobile agents are software nomads that act as your personal representative, working autonomously through networks. They are able to visit network nodes directly using available computing power and are not limited by platform. This emerging field is now poised to become a cornerstone for new Web-based ubiquitous computing environments. Mobile Agents provides a practical introduction to mobile agent technology and surveys the state of the art in mobile agent research. Students and researchers can use the book as an introduction to the concepts and possibilities of this field and as an overview of ongoing research. Developers can use it to identify the capabilities of the technology to decide if mobile agents are the right solution for them. Practitioners can also gain hands-on experience in programming mobile agents through exploration of the source code for a complete mobile agent environment available through the companion website.

2. Mobile Agents Objective

The mobile agent paradigm is applicable in many application fields from which some areas are specially recognized as e-commerce, electronic auctions and stock market. These areas need high transaction of money. Now the user will put his money in mobile-agent only when he is confident that his money is secure and the agents can be trusted which are dealing with his money or transmitting some secure information. This is one reason which investigates much research effort in security of mobile-agents which has its own benefits when used in these areas. As the name suggests that these agents are mobile in nature, which make them to travel autonomously in the network, due to this nature the agents become more vulnerable to various attacks. These attacks expose the limitation of mobile agents in the area of security. Until all or most of these attacks are resolved the usage of mobile-agents to its full potential will remain a constraint.

3. Mobile Agents Properties

Mobile agents have the following unique properties:

- Adaptive Learning:** Mobile agents can learn from experiences and adapt themselves to the environment. And can monitor traffic in large networks and learn about the trouble spots in the network. Based on the experiences of the agent in the network the agent can choose better routes to reach the next host.
- Autonomy:** Mobile agents can take some decisions on its own. For example, mobile agents are free to choose the next host and when to migrate to the next host. These decisions are transparent to the user and the decisions are taken in the interest of the user.
- Mobility:** Mobile agents have the ability to move from one host to another in the network.

4. Mobile Agent Architecture

The architecture gives the structure of the system which consists of some components, their individual functionalities and their interrelationship with each other. The basic architecture of the mobile agent can be thought of as a client sends out an agent who travels the network visiting servers in order to perform some required action. The architecture consists of:

A. Agent Manager: Agent manager has few responsibilities as it:

- Sends agents to and receives agents from remote hosts.
- Prepares agents for transport by serializing the agent.
- Reconstructs received agents and creates the agents execution context.

B. Security Manager: Responsibilities of security manager are:

- Authenticates agents before allowing execution.

Volume 5 Issue 10, October 2016

www.ijsr.net

[Licensed Under Creative Commons Attribution CC BY](https://creativecommons.org/licenses/by/4.0/)

- Automatically invoked when the agents tries to use any system resource or tries for any unauthorized activity.
- Protects the host and agent from unauthorized access.

C. Directory Manager: Lists names and addresses of services and agents as shown in Fig. shows the agent first migrates to remote container and registers itself to the Directory Services. When any other agent needs to find some agent it contacts the Directory Services for help.

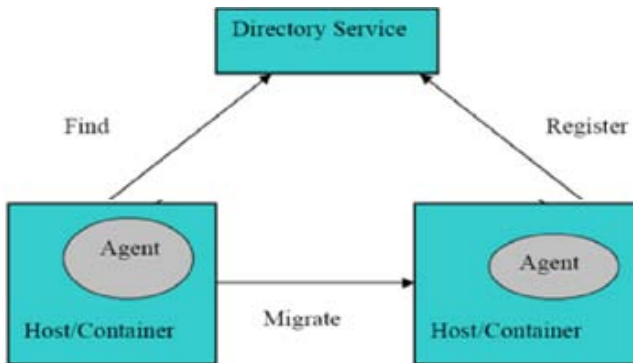


Figure: Mobile Agent Architecture

D. Language: The architecture of a mobile agent system describes the flow of information among the various components of the system. The language used for the efficient transfer of data and provides the developer with the tools to efficiently implements the system. Most current agent systems are implemented on top of the Java Virtual Machine (JVM), which provides object serialization and basic mechanism to implement weak mobility.

E. Agent Platform: Mobile agents are the basis of the emerging technology which makes it easier to design, implement and maintain distributed systems. Mobile agents have the unique ability to transport themselves from one system in a network to another. We found that mobile agents reduce network traffic, overcome network latency and most importantly their ability to operate asynchronously and autonomously. It helps us to construct more robust and fault tolerant systems. Aglets are Java objects that can move from one host on the internet to another. That is, an aglet can execute on one host, can stop its execution, dispatches itself to other host and resume its execution on new remote host. On moving from one system to another aglet carries its code and data with it. The word aglet means “lightweight agent” in much the same way that applet means lightweight application. Aglets are hosted by an aglet server in a similar way in which an applet is hosted by a web browser. The Aglet server provides an environment where agents can execute and Java language and Aglet security manager make the agents transfer safe. The Aglets Software Development

Kit (ASDK) is an implementation of an Aglet API. The ASDK includes Aglet API packages, documentation, sample aglets, and the Tahiti Server. Tahiti is java application that allows the users to receive, manage, and send aglets to other computers that are running Tahiti.

5. Mobile Java Agent: The Aglet Model

The aglet model allows the proper and easier use of the agents created. The aglet model helps in setting up proper infrastructure through which we can use and take advantage of the agents.

Basic Elements

The aglet object model explains some abstraction and the behaviour which is used to take full advantage of this agent technology. The abstractions which are used are:

- **Aglet:** An aglet is a java object which moves in a network and gets executes on host which are aglet enabled. It is autonomous and run in its own thread. Proxy: a proxy is a representative of an aglet. It also protects the aglet from direct access to its public methods. The proxy also provides the location transparency for the aglet.
- **Context:** A context is where an aglet executes. It is a stationary object that provides a means for maintaining and managing aglets.
- **Identifier:** An identifier is bound to an aglet. This identifier is globally unique an immutable throughout the lifetime of the aglet.
- **Creation:** The creation of aglets takes place in a Context. The new aglet is assigned an identifier, inserted into the context, and initialized.
- **Cloning:** The cloning produces exact copy of the original aglet in the same context. The cloned aglet has different identifier
- **Dispatching:** Dispatching an aglet from one context to another will remove it from its current context and insert into the destination context, where it will restart execution. This process is termed as dispatching.
- **Retraction:** The retraction will pull aglet from its current context and insert it into the context from which the retraction was requested.
- **Activation and Deactivation:** the deactivation of an aglet will halt its execution for the mentioned amount of time and store its state in secondary storage. Activation will again restore it in the same context.
- **Disposal:** The disposal of an aglet will halt its current execution and remove it from its current context.

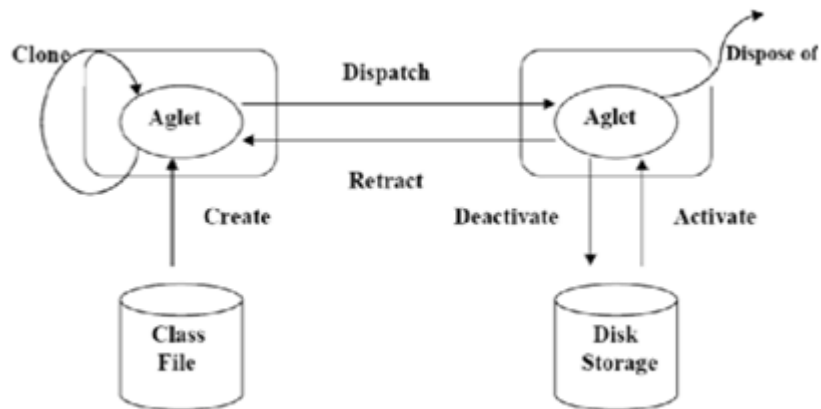


Figure: Fundamental operations of Aglet

6. Mobile Agents Hurdles

1) Technical Hurdles

a) Performance and Scalability.

Since agents are often written in a slow interpreted language for portability and security reasons, and since the agents must be injected into an appropriate execution environment upon arrival. Thus, in the absence of network disconnections, mobile agents often take longer to accomplish a task than more traditional implementations, since the time savings from avoiding intermediate network traffic is currently less than the time penalties from slower execution and the migration overhead.

b) Portability and standardization

For mobile agents code must be portable across mobile-code systems, since it is unreasonable to expect that the computing community will settle on a single mobile-code system. Making code portable across systems will require a significant standardization effort, the next step of standardizing on some specific execution environment(s) such as a particular virtual machine, as well as on the format in which the code and state of a migrating agent are encoded.

c) Security

These three problems will severely limit the use of mobile agents in a truly open environment such as the Internet :(1) protecting the machines without artificially limiting agent access rights, (2) protecting an agent from malicious machines; and (3) protecting groups of machines that are not under single administrative control. Although many technical advances (and user-education efforts) must be made before these three problems are solved adequately for all Internet applications, current work is promising enough that, within a few years, mobile-agent systems will be secure enough for many applications.

2) Non-technical hurdles

a) Lack of a killer application

The most important hurdle is that there is no killer application for mobile agents. Thus, the advantages of mobile agents are modest when any particular application is considered in isolation. Instead a set of applications and argue that the entire set can be implemented with much

less effort.

b) Getting ahead of the evolutionary path.

Full mobile-agent systems involve all the same research issues (and more) as more restricted mobile-code systems, researchers must be careful to demonstrate that the switch to mobile agents can be made incrementally. For example, applets, mobile code that migrates from server to client for better interaction with the user, are in common use, and the associated commercial technology is improving rapidly (e.g., faster Java virtual machines with just-in-time compilation). From applets, the next step is proxy sites that accept mobile code sent from a mobile client. In all likelihood, such proxies will be first provided by existing Internet service providers (ISPs). Since the sole function of the proxy sites will be to host mobile code, and since the ISPs will receive direct payment for the proxy service (in the form of user subscriptions, although not likely at a fixed rate), the ISPs will be willing to accept the perceived security risks of mobile code. Once mobile-code security is further tested on proxy sites, the services themselves will start to accept “servlets”, mobile code sent from the client directly to the server (or from the proxy to the server).

c) Revenue and image

A final important hurdle is the problem of revenue flow and commercial image. For example, although it is not yet clear whether advertising is a viable economic foundation for Web sites, many Web sites earn money solely from advertisements. If these sites allow mobile agents to easily access the content of the site, the number of human visits to the Web pages will presumably decrease, and the advertisements will not be seen. A poorly implemented agent may lead to a negative view of the service, even though the service is blameless.

7. Mobile Agents Pros & Cons

Pros:

- Mobile Agents is the collecting and processing of information. The collecting of information is still done in a very centralized way.
- It works locally on a system and not to be dependent on a connection to this system.
- This can improve the monitoring of systems or processes, which can be done locally on a machine.

- Mobile Agents is the smaller use of bandwidth or, more generally, telecommunication infrastructure. Since the backbone of the internet is getting improved in a very fast way the access methods for the end user are still limited to a small bandwidth (most 56kbps through analog modems).

Cons:-

The main drawback of mobile agents security risk mobile computing environment are twofold: Firstly a malicious mobile agent can damage a host. On the other hand a malicious host can tamper with the functioning of the mobile agent. The defense mechanisms suggested try to prevent malicious actions in the first place. These include safe programming Languages that prevent mobile agents from performing malicious actions on the hosts. If a malicious action does occur then the defense mechanisms detect it as soon as possible and take remedial action. One of the schemes proposed is to introduce a tracing mechanism that records the execution of the mobile agent at each host. When the agent is dispatched to the next host, the trace is also sent. Using this trace, malicious actions can be detected and the malicious host can be identified.

8. Future Scope

The area of mobile agent applications is still in a somewhat immature state. The traditional host orientation toward security persists, and the focus of protection mechanisms within the mobile agent paradigm remains on protecting the agent platform. However, emphasis is moving toward developing techniques that are directed towards protecting the agent, a much more difficult problem. Fortunately, there are a number of applications for agents where conventional and recently introduced security techniques should prove adequate, until further progress can be made. The next wave of security improvements for agent systems is likely to emerge from the present baseline of protection techniques, either through incremental refinements that reduce processing and storage overhead or simplify the use of the mechanism, or clever combination of complementary mechanisms to form a more effective composite protection scheme. An opportunity for users along the following lines:

- Agent Security Framework.
- Security Design Tools.
- PKI Privilege Management Extensions.
- Anonymity.

9. Conclusion

This present work examined the scope of the focusing on used architecture and agent platforms for the mobile. In this paper I discussed about the mobile-agents, its security issues and its countermeasures. Mobile agent system is a very promising paradigm; it has shown its presence in many applications like distributed networking, e-commerce applications etc. There are numerous advantages of using the mobile agent paradigm rather than conventional paradigm such as client-server based technologies. In one way it provides the abstraction to networking. The benefits of mobile-agent technology cannot be exploited fully until all the security issues are properly addressed. I try to provide a better solution by combining the results of two or more

solutions. This solution addresses most of the problem but still it is very much dependent on the complexity of the architecture used and the possibility.

References

- [1] Mobile Agents White Paper," General Magic, 1998.Future Directions for Mobile Agent Research David Kotz, Robert Gray, and Daniela Rus Dartmouth College.
- [2] James Gosling and Henry McGilton, "The Java Language Environment: A White Paper," Sun Microsystems, May 1996.
- [3] Danny Lange and Mitsuru Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley, 1998.
- [4] Tiwari, Rajnish and Buse, Stephan (2007): The Mobile Commerce Prospects: A Strategic Analysis of Opportunities in the Banking Sector, Hamburg University Press (E-Book as PDF to be downloaded).
- [5] Tiwari, Rajnish; Buse, Stephan and Herstatt, Cornelius (2007).
- [6] Object Management Group, NEEDAM MA, Agent Technology Green Paper 1 Sept 2000, PP. 8-11.
- [7] The Foundation for Intelligent Physical Agents (FIPA), <http://www.fipa.org/>
- [8] Hyacinth S. Nwana, "Software Agents: An Overview", Intelligent systems Research, Advanced Applications & Technology Department, Ipswich, Suffolk U.K. Cambridge University Press 1996.
- [9] Colin G. Harrison, David M. Chess, and Aaron Kershenbaum, "Mobile Agents: Are they a good idea?", technical report, 1995, IBM Research Division.
- [10] W. Jansen and T. Karygiannis, "Mobile Agent Security", Nist Special Publication 800-19 -, 2000. National Institute of Standards Technology.